1. The Combination of Richburg and Kim

The Office Action asserts, "it would be obvious for a skill in the art at the time the invention was made to add to Richburg's software product instructions to perform the mapping of functional requirements against design parameters with matrix as taught by Kim. The motivation would be obvious since having included in the framework program capability to provide a visual structure like matrix to verify the correctness of what to implement versus predetermined functional requirements would enhance considerable the stages of designing, implementing, and verifying software product." *See* Office Action, page 4. Applicants respectfully disagree with this assertion.

One of skill in the art would not have been motivated to incorporate a design matrix disclosed by Kim into the software product of Richburg. Kim discloses that design matrices are useful in creating solutions to problems. Richburg, however, is not related to creating solutions to problems, but rather relates to the manipulation of solutions that have already been created. That is, Richburg relates to taking a solution that has already been created and is stored in the knowledgebase and converting that solution from the format of the information in the knowledgebase into software code (*see* col. 7, lines 53-68 and col. 5, lines 4-11). Richburg compares the information stored in the knowledgebase to a recipe used in cooking, which describes the detailed steps in a cooking process and identifies specific ingredients, amounts, and variations in meal preparation procedures for making a particular dish (col. 9, lines 26-30). Thus, like a cooking recipe, the knowledgebase files represent a solution to a problem. Richburg discloses a method for translating this solution into software code.

This is different from the disclosure of Kim, wherein a design matrix may be used in the axiomatic design process to develop or design a solution to a problem. For example, Kim discloses that the design of a software system consists of an FR (functional requirement) hierarchy in the functional domain and a DP (design parameter) hierarchy in the physical domain. FRs are the outputs of the software and DPs are the key inputs to the software which can characterize and control the FRs. The software code is represented by a set of design matrices which transform DPs to FRs at each level of the hierarchy. Therefore, the generation of the hierarchical trees of FRs and DPs constitutes the design process (Kim, page 246).

In Richburg, a solution has already been designed and is stored in the knowledgebase files. Richburg relates to generating software code from the previously created knowledgebase files. Because the design matrix of Kim is used to design a solution, such a design matrix would serve no purpose in the system of Richburg. Thus, one of skill in the art would not have been motivated to incorporate the design matrix of Kim into the system of Richburg, as the Office Action asserts. Accordingly, it is respectfully requested that the combination of Richburg and Kim be withdrawn.

2. Applicants' Claims Patentably Distinguish Over Richburg and Kim

Even if one were to combine Richburg and Kim, Applicants' claims patentably distinguish over any such combination.

Claim 1

Claim 1 is directed to a digital information product comprising: a computer-readable medium; and stored on the computer-readable medium, computer program instructions defining a software system that produces software code based on a set of functional requirements and design parameters provided by a programmer, wherein the computer program instructions, when executed, allow the programmer to define a design matrix describing a relation between the set of functional requirements and the design parameters.

The Office Action concedes that Richburg does not disclose the use of functional requirements, but asserts, "[t]he concept of translating functional requirements from customer needs and fulfilling them with implementation in a software or other designs use CASE tool/mddleware (e.g., Rationale Rose/UML) was a known concept at the time the invention was made; and establishing or mapping those functional requirements against the design specification or parameters of a target application was a known concept..." *See* Office Action, page 3. Applicants traverse any assertion that these concepts were known at the time of the invention. If the rejection is to be maintained, the Examiner is respectfully requested to cite a reference in support of this assertion.

The Office Action further concedes that Richburg fails to disclose program instructions that allow the programmer to define a design matrix describing a relation between the set of functional requirements and the design parameters, but asserts, "the concept of mapping requirements against all functionalities of a target software application, e.g., validation spreadsheet or traceability matrix in Modeling tool, was a known concept at the time the invention was made..." *See* Office Action, page 3. Applicants traverse any assertion that this concept was known. If the rejection is to be maintained, Applicants respectfully request that the Examiner cite a reference in support of this assertion.

Further, neither Richburg or Kim, taken alone or in combination, discloses or suggests, "computer program instructions defining a software system that produces software code based on a set of functional requirements and design parameters provided by a programmer."

While Kim discloses using functional requirements and design parameters to define a software system, Kim does not disclose a software system that produces software code based on these functional requirements and design parameters. Richburg also fails to disclose a software system that produces software code based on a set of functional requirements and design parameters, as the software system of Richburg generates code based on knowledgebase files.

Thus, claim 1 patentably distinguishes over Richburg and Kim, taken alone or in combination. Accordingly, it is respectfully requested that the rejection of claim 1 under 35 U.S.C. §103(a) be withdrawn.

Claims 3-5 depend from claim 1 and are patentable for at least the same reasons. Accordingly, it is respectfully requested that the rejection of claims 3-5 under 35 U.S.C. §103(a) be withdrawn.

<u>Claim 64</u>

Claim 64 is directed to a database format for designing a software system comprising: design identification information identifying information describing the software system; detailed design description information describing the structure and operating qualities of the software system, wherein the detailed design description information defines a design matrix describing a relation between a plurality of functional requirements of the software system and design parameters, represents at least one software object by at least one of the functional requirements, and represents data used by the at least one software object by at least one of the design parameters; and software code information associated with the software system.

The Office Action concedes that Richburg fails to disclose detailed design information describing the structure and operating qualities of the software system, but asserts that Richburg implicitly discloses this limitation. *See* Office Action, pages 4-5. Applicants respectfully disagree. For this limitation to be present in Richburg, the limitation must either be disclosed by Richburg or must be inherent. The Office Action concedes that this limitation is not disclosed by Richburg. Further, this limitation is not inherent. MPEP §2112 states that, "[t]o establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.'" MPEP, 8th Edition, Rev. 2, May 2004, page 2100-54. The Office Action fails to establish that this limitation is necessarily present in Richburg.

Further, neither Richburg nor Kim, taken alone or combination, discloses or suggests that, "the detailed design description information defines a design matrix describing a relation between a plurality of functional requirements of the software system and design parameters, represents at least one software object by at least one of the functional requirements, and represents data used by the at least one software object by at least one of the design parameters," as recited in claim 64.

The Office Action concedes that Richburg fails to disclose or suggest this limitation. *See* Office Action, page 5. Kim does not cure this infirmity of Richburg. Kim discloses a design matrix that describes a relation between functional requirements and design parameters, but does not disclose or suggest that at least one of the functional requirements represents at least one software object and at least one of the design parameters represents data used by the software object.

Thus, claim 64 patentably distinguishes over Richburg and Kim, taken alone or in combination. Accordingly, it is respectfully requested that the rejection of claim 64 under 35 U.S.C. §103(a) be withdrawn.

Claim 66 depends from claim 64 and is patentable for at least the same reasons. Accordingly, it is respectfully requested that the rejection of claim 66 under 35 U.S.C. §103(a) be withdrawn.

B. Claims 6, 15, 31, 36, 48, 67, and 86

The Office Action rejected claims 6-13, 15, 17-51, 53-63, 67-94 under 35 U.S.C. 103(a) as purportedly being obvious over Richburg in view of Underwood (6,523,027) and Rudolph ("On A Mathematical Foundation of Axiomatic Design," August 22, 1996, ASME Design Engineering Technical Conference and Computers in Engineering Conference). Applicants respectfully traverse each of these rejections, as the combination of Richburg, Underwood, and Rudolph is improper and, even if one were to combine the references, Applicants' claims patentably distinguish over any such combination.

1. The Combination of Richburg , Rudolph, and Underwood

In Applicants' response of February 17, 2004, Applicants argued that none of the cited references discloses or suggests applying the axiomatic design principles disclosed by Rudolph to the design of software code and the automatic generation of software code. Applicants further argued that neither Richburg nor Underwood even mentions axiomatic design and that the only suggestion of applying axiomatic design principles to the design of software comes from Applicants' own disclosure.

In response to this argument the Office Action asserts that, "The rationale as to use a matrix for mapping has been address in the rejection, not only because this feature is well known in the art of design and engineering but also because both Underwood and Richburg's inventions are purported to fulfill translating of user's derived required features into deliverables with a similar modeling tool." *See* Office Action, page 26. Applicants respectfully disagree with this assertion. Initially, Applicants traverse any assertion that the use of a matrix in the design of software was well-known in the art at the time of the invention. If the rejection is to be maintained, the Examiner is respectfully requested to cite a reference in support of this assertion.

Additionally, neither Underwood nor Richburg discloses or suggests the use of a matrix in software design. Indeed, **Underwood is completely unrelated to the design of software**. While Underwood does disclose the use of matrices for other purposes, these matrices are not used in the design of software or generation of software code. Richburg also fails to disclose or suggest the use of a matrix in the design of software. In Richburg, software is already designed and stored in the knowledgebase as knowledgebase files. Richburg merely discloses how code may be generated based on these knowledgebase files.

Thus, there is nothing in any of the references that would have motivated one of skill in the art to apply the design methodology of Rudolph to the design of software. Accordingly, one of skill in the art would not have combined Richburg, Underwood, and Rudolph in the manner suggested in the Office Action. Thus, it is respectfully requested that any rejection based on such a combination be withdrawn.

2. Applicants' Claims Patentably Distinguish Over Richburg , Rudolph, and Underwood

Even if one were to combine the references in the manner suggested in the Office Action, Applicants' claims patentably distinguish over any such combination.

None of the references discloses or suggests generating software code based on a design matrix. While Richburg discloses automatically generating software code, this software code is generated based on knowledgebase files. Generating software code from a design matrix is different from generating software code from knowledgebase files. There is nothing in any of the references that suggests using a design matrix to generate software code or how this would even be done.

Claim 6

Claim 6 is directed to a method for producing a software system. The method comprises: "defining a design matrix describing a relation between functional requirements of a software system and design parameters; and generating software code based on the design matrix."

As should be clear from the discussion above, claim 6 patentably distinguishes over Richburg, Underwood, and Rudolph, whether taken alone or in combination, as none of the references discloses or suggests generating software code based on a design matrix. Accordingly, withdrawal of the rejection of claim 6 under 35 U.S.C. §103(a) is respectfully requested.

Claims 7-14 depend from claim 6 and are patentable for at least the same reasons. Accordingly, withdrawal of the rejection of claims 7-14 is respectfully requested.

Claim 15

Claim 15 is directed to a method for producing software. The method comprises: defining a design matrix describing a relation between functional requirements of a software system and design parameters so that the design matrix has a lower triangular form; and generating software code based on the design matrix.

As should be clear from the discussion above, claim 15 patentably distinguishes over Richburg, Underwood, and Rudolph, whether taken alone or in combination, as none of the references discloses or suggests generating software code based on a design matrix. Accordingly, withdrawal of the rejection of claim 15 under 35 U.S.C. §103(a) is respectfully requested.

Claims 17-30 depend from claim 15 and are patentable for at least the same reasons. Accordingly, withdrawal of the rejection of claims 17-30 is respectfully requested.

Claim 31

Claim 31 is directed to a method for designing software involving object-oriented software objects. The method comprises: defining a design matrix describing a relation between a plurality of functional requirements of the software system and design parameters; representing at least one object-oriented object by at least one of the functional requirements; representing data used by the at least one object-oriented software object by at least one of the design parameters; and representing a method of the at least one object-oriented software object by a product of a portion of the design matrix and the at least one design parameter.

As should be clear from the discussion above, claim 31 patentably distinguishes over Richburg, Underwood, and Rudolph, whether taken alone or in combination, as none of the references discloses or suggests representing a method of the at least one object-oriented software object by a product of a portion of the design matrix and the at least one design parameter. Accordingly, withdrawal of the rejection of claim 31 under 35 U.S.C. §103(a) is respectfully requested.

Claims 32-35 depend from claim 31 and are patentable for at least the same reasons. Accordingly, withdrawal of the rejection of claims 32-35 is respectfully requested.

Claim 36

Claim 36 is directed to a method for designing software comprising: defining a software system by defining a design matrix describing a relation between functional requirements of the software system and design parameters implementing the software system; defining at least one object-oriented object; and defining at least one method that may be defined on the at least one object, wherein the at least one object-oriented object and the at least one method are related to the design matrix.

As should be clear from the discussion above, claim 36 patentably distinguishes over Richburg, Underwood, and Rudolph, whether taken alone or in combination, as none of the references discloses or suggests defining a design matrix describing a relation between functional requirements of the software system and design parameters implementing the software system. Accordingly, withdrawal of the rejection of claim 36 under 35 U.S.C. §103(a) is respectfully requested.

Claims 37-47 depend from claim 36 and are patentable for at least the same reasons. Accordingly, withdrawal of the rejection of claims 37-47 is respectfully requested.

Claim 48

Claim 48 is directed to a database format for designing a software system comprising: a software design specification, wherein the design specification defines a design matrix describing a relation between a plurality of functional requirements of the software system and design parameters, wherein the design specification represents at least one software object by at least one of the functional requirements, and wherein the design specification represents data used by the at least one software object by at least one of the design parameters; and software code produced by the design specifications.

As should be clear from the discussion above, claim 48 patentably distinguishes over Richburg, Underwood, and Rudolph, whether taken alone or in combination, as none of the references discloses or suggests a design specification that defines a design matrix describing a relation between a plurality of functional requirements of the software system and design parameters. Accordingly, withdrawal of the rejection of claim 48 under 35 U.S.C. §103(a) is respectfully requested.

Claims 48-51 and 53-63 depend from claim 48 and are patentable for at least the same reasons. Accordingly, withdrawal of the rejection of claims 48-51 and 53-63 is respectfully requested.

Claim 67

Claim 67 is directed to a method for generating software code associated with a software system. The method comprises: defining a design matrix describing a relation between functional requirements of the software system and design parameters; and generating software code based on the design matrix.

As should be clear from the discussion above, claim 67 patentably distinguishes over Richburg, Underwood, and Rudolph, whether taken alone or in combination, as none of the references discloses or suggests a defining a design matrix describing a relation between functional requirements of a software system and design parameters. Accordingly, withdrawal of the rejection of claim 67 under 35 U.S.C. §103(a) is respectfully requested.

Claims 68-85 depend from claim 67 and are patentable for at least the same reasons. Accordingly, withdrawal of the rejection of claims 68-85 is respectfully requested.

Claim 86

Claim 86 is directed to a computer-readable medium encoded with a program that, when executed on a computer system, performs a method for rendering an interface through which a user may interact. The method comprises steps of: displaying a software design interface, wherein the interface displays a set of functional requirements upon which a software design is based, and wherein the interface displays a design matrix describing a relation between the set of functional requirements and design parameters implementing the software design.

As should be clear from the discussion above, claim 86 patentably distinguishes over Richburg, Underwood, and Rudolph, whether taken alone or in combination, as none of the references discloses or suggests an interface that displays a design matrix describing a relation between the set of functional requirements and design parameters implementing the software design. Accordingly, withdrawal of the rejection of claim 86 under 35 U.S.C. §103(a) is respectfully requested.

Claims 87-95 depend from claim 86 and are patentable for at least the same reasons. Accordingly, withdrawal of the rejection of claims 87-95 is respectfully requested.

CONCLUSION

In view of the foregoing remarks, this application should now be in condition for allowance. A notice to this effect is respectfully requested. If the Examiner believes, after this request for reconsideration, that the application is not in condition for allowance, the Examiner is requested to call the Applicant's attorney at the telephone number listed below.

If this response is not considered timely filed and if a request for an extension of time is otherwise absent, Applicant hereby requests any necessary extension of time. If there is a fee occasioned by this response, including an extension fee, that is not covered by an enclosed check, please charge any deficiency to Deposit Account No. 23/2825.

Respectfully Submitted,

Sung-Hee Do et al., Applicants

By:

Richard F. Giunta, Reg. No. 36,149
WOLF, GREENFIELD & SACKS, P.C.
600 Atlantic Avenue
Boston, Massachusetts 02210-2211
Telephone: (617) 720-3500
Attorneys for Applicants

Docket No.: A0734.70001US00
Date: November 3, 2004
x11/04/04x